

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 0 756 228 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
02.09.1998 Bulletin 1998/36

(51) Int Cl.⁶: G06F 3/06, G06F 9/46

(21) Application number: 96650025.8

(22) Date of filing: 01.07.1996

(54) **Augmenting volume management of information storage devices to handle direct access to storage devices**

Erweitern der Datenträgerverwaltung von Informationsspeichereinrichtungen zum Behandeln des direkten Zugriffs zu Speichereinrichtungen

Expansion de la gestion des volumes de dispositifs de mémorisation d'informations pour traiter des accès directs aux dispositifs de mémorisation

(84) Designated Contracting States:
DE FR GB IE IT SE

(30) Priority: 06.07.1995 US 498996

(43) Date of publication of application:
29.01.1997 Bulletin 1997/05

(73) Proprietor: SUN MICROSYSTEMS, INC.
Mountain View, California 94043-1100 (US)

(72) Inventors:
• Duncan, William W.
Colorado Springs, Colorado 80919 (US)
• Alt, Howard
Colorado Springs, Colorado 80919 (US)

(74) Representative: Hanna, Peter William Derek et al
Tomkins & Co.,
5 Dartmouth Road
Dublin 6 (IE)

(56) References cited:
EP-A- 0 482 853 WO-A-91/19246
US-A- 4 819 159

- IBM TECHNICAL DISCLOSURE BULLETIN, vol. 34, no. 10A, 1 March 1992, pages 124-125, XP000302249 "POLL AND SELECT FUNCTION FOR CASCADED DEVICE DRIVERS ON A UNIX OPERATING SYSTEM"

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 756 228 B1

Description

Background of the Invention

Field of the Invention

This invention relates to a volume management system wherein the system using the volume management driver may also access a storage device directly without conflicting with the volume management system.

More particularly, the invention relates to augmenting the volume management driver in a UNIX operating system to add a capability to operate in a manner that allows the user to issue direct commands to a device through the volume management system drivers.

Description of Related Art:

In a UNIX operating system there exist volume management systems for devices particularly disk drive devices for managing the storage of information on optical or magnetic disk in a storage system. The volume management software is part of the kernel in the UNIX operating system. The objective of this volume management software is to manage the storage media as opposed to management of the storage device. For example, the volume management software would access storage media by name such as the volume name for a floppy disk in a diskette drive or for cylinders in a hard disk drive. Before volume management was available, a device driver in the UNIX kernel provided direct access to a device by the device name as for example, the A disk drive or the C disk drive.

The present design of volume management system drivers is referred to as layered. As an example of operating in a layered volume management system, Fig. 1 shows a read volume operation. Operation 11 from the application program (user) to the operating system kernel is "read /vol," i.e. a request to read data from a location identified as a storage media volume. The process flows from the read request operation 11 from the user to the kernel of the UNIX operating system where a device driver at a first layer in the volume management system is the block device module "bdev_strategy (vol_dev)" 12. Device driver 12 at this first layer interprets the read operation for the volume and identifies the type (storage, printer, display, etc.) of device as specified by the "(vol_dev)" parameters. The volume parameters in this case identify a storage type of device. This first driver routine 12 calls module 14, "volstrategy (vol_dev)," at the next layer. Routine 12 knows that for vol_dev parameters received, it needs to call volstrategy (vol_dev). Module 14 further interprets vol_dev parameters received from module 12 and identifies the storage device or storage system where the volume is located. The volstrategy module 14 then drives routine 16 back up at the device driver layer 12. Module 14 identifies the

vol_dev number received as actual device number "lp->vol_dev," so it passes that to the block device driver routine 16, "bdev_strategy(lp->vol_dev)." Routine 16 identifies a type of storage device, a hard disk drive, a floppy disk drive or yet another storage system. In this example the device is a floppy disk drive or diskette drive. Routine 16 calls module 18 which is the floppy disk device driver for the diskette drive where the volume is located. Module 18 is the lowest layered level, and in this example module 18 is "fdstrategy(fd_dev)," a floppy disk driver routine.

In this layered volume management system the device that is ultimately opened and contains the volume being accessed is exclusively opened by the volume management system. Therefore, it is not possible for a direct access read operation such as "read /device" operation 19 from the user to gain access to a device already accessed by the volume management system. This is indicated in Fig. 1 by the blocked path between operation 19 and module 18.

In the past users were not able to get direct access to a device, and were only able to access it through the volume management system, but first they had to go through a volume check command and then recite a path from the volume down to the device where the information in the volume to be accessed was located. This level of complexity for the user to be able to access a device has prompted many users just to turn off volume management and use only device direct access commands. What is needed is the ability to have the UNIX operating system interpret both a volume management access command from the user and a direct device access command from the user without conflict.

Direct access to a device through multiple layers of drivers is typically referred to in the art as stacked device drivers. In stacked device drivers the intermediate drivers are transparent. Accordingly, the read command from the user may proceed down the device driver layer and into the storage device driver layer without any volume path specification by the user. What is needed by the user might be referred to as a volume management system that is both layered and stacked.

The following background art relates in general to the problem of device or volume access conflicts in a UNIX environment.

For example, EP-A-0,482,853 is directed to a volume management system in a UNIX environment. More particularly, it relates to maintaining a volume group status area to track updates to volumes in a volume group. However, it does not teach or suggest managing the combination of an open volume operative system and an open device operative system, as described and claimed in the present invention.

IBM TECHNICAL DISCLOSURE BULLETIN, Vol. 34, No. 10A, March 1, 1992, pages 124-125 discloses a "mechanism to allow cascaded device drivers to pass along select/poll I/O event requests."

WO-A-91 19246 includes a description of a process

for opening and closing a disk drive device.

US-A-4,819,159 relates to a fault tolerant system building block (SBB) or manager where if there is a failure by one such manager in control ("ownership") of a disk, then a second SBB takes over control of the disk.

This background art does not contain a teaching of the management of both volume access requests and device access requests as described and claimed in the present invention.

Summary of the Invention

The present invention relates to apparatus and a method for managing the access to storage devices by user requests to a computing system, in accordance with claims 1 and 8, which follow.

It is an object of this invention to provide a layered and stacked volume management system.

In accordance with this invention, the above object is accomplished by a computing system operating a volume management system for the storage of information and providing to the users of the volume management system parallel process paths for accessing a storage device as an access volume operation or as an access device operation. Further the volume management system prevents the two parallel logical operations from conflicting with each other by performing open volume operation and open device operation that indirectly communicate by setting and clearing volume data. The volume data is meta-data that describes characteristics of the volume being accessed and the device being accessed. The meta-data includes an open count and an exclusive flag to track the open status of a device. Open count indicates if the device is open, and the exclusive flag indicates whether the storage device has been opened exclusively by either the open volume or the open device operation request from a user.

The inventive process includes computer implemented steps of accessing a storage device through parallel processes -- access volume process and access device process. The access volume process and the access device process both detect whether the requested storage device is already open and whether or not its opened exclusively. This detection is accomplished by testing for an exclusive flag in the volume data for the device the user requested access to. Next the access volume process and the access device process test whether the request is an exclusive open request. These tests allow the parallel processes to maintain an open count and an exclusive flag for each storage device. By monitoring the open count for non-zero and the exclusive flag, each process can return a busy code when the requested device is already exclusively open or the request is for an exclusive open, and the device has already been opened by a previous request that is still pending, i.e. has not been closed.

The great advantage of this invention is that it permits a user to access storage devices either through an

access volume operation or an access device operation. To the user either process is equally available and thus the user can operate both processes in parallel without concern for disabling or inhibiting the volume management system.

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of a preferred embodiment of the invention as illustrated in the accompanying drawings.

Brief Description of Drawings

Fig. 1 is a flow diagram of the layered volume management system as it exists in the prior art.

Fig. 2 is a flow diagram of a preferred embodiment of the invention, implementing both the layered and stacked volume management system to perform a read operation.

Fig. 3 illustrates a computing system to perform the computer implemented steps in accordance with the invention.

Fig. 4 shows a preferred embodiment for the open volume and open device processes and logical operations of the invention.

Fig. 5 shows the structure of the volume data indicating the open count and exclusive flag in the meta data making up the volume data.

Fig. 6 shows the detailed operation flow of the volume open process in Fig. 4.

Fig. 7 shows the detailed operation flow of the device open process in Fig. 4.

Fig. 8 shows a preferred embodiment of the volume close and device close processes in accordance with the invention.

Detailed Description of Preferred Embodiments

The volume management system operations involve four primary operations -- open, read, write, and close. Each of these operations follow the parallel flow through layers of driver modules. In one process path, the layers are access volume process defined. In the other process path, the layers are access device process defined. The parallel process paths for read and write operations are independent process flows. However, the parallel process paths for the open operation and the close operation must interact to deal with accesses that require exclusive use of the device.

As shown in Fig. 2, a preferred embodiment of the layered and stacked volume management system for the read operation is initiated by the user through either a read from volume operation 20 or a read from device operation 22. The read from volume operation 20 request is passed to the device driver "bdev_strategy (vol_dev)" module 24. In turn, the block device strategy module 24 calls a volume strategy module 26 based on the type of device identified in the vol_dev parameters.

The volume strategy driver 26 calls a device driver 30 based on the device identified by the vol_dev. Block device strategy module 30 then calls the actual device strategy driver, which in this case is a floppy disk device driver module 32.

In the process flow for the read from device operation 22 from the user, operation 22 calls the block device strategy module 34 for a floppy disk device. This driver module 34 then drives a volume strategy driver module 28 which calls the floppy disk strategy device driver module 32. In this way, the layered process 24, 26, 30 and stacked process 34, 28 for the volume management system operate in parallel. The write operation flow is identical to the read operation flow.

The operating environment in which the present invention is used encompasses the general distributed computing system, wherein general purpose computers, workstations, or personal computers are connected via communication links of various types, in a client-server arrangement, wherein programs and data, many in the form of objects, are made available by various members of the system. Some of the elements of a general purpose workstation computer are shown in Fig. 3, wherein a processor 1 is shown, having an input/output (I/O) section 2, a central processing unit (CPU) 3 and a memory section 4. The I/O section 2 is connected to a keyboard 5, a display unit 6, a disk storage unit 9 and a CD-ROM drive unit 7. The CD-ROM unit 7 can read a CD-ROM medium 8 which typically contains programs 10 and data. The computer program products containing mechanisms to effectuate the apparatus and methods of the present invention may reside in the memory section 4, or on a disk storage unit 9, or on the CD-ROM 8 of such a system. Examples of such systems include SPARC systems offered by Sun Microsystems, Inc., personal computers offered by IBM Corporation and by other manufacturers of IBM-compatible personal computers, and systems running the UNIX operating system.

Before an access (read or write) operation may be performed as described above with reference to Fig. 2 possible conflicts in access to the same storage device by either the read from volume operation 20 or the read from device operation 22 must be resolved by an open operation. The open operation as shown in Fig. 4 begins with either a user open volume request operation 36 of "open of device in /vol" or a user open device request operation 38 of "open of device in /dev." The volume open operation calls the "dev_open(vol_path)" driver module 40. Dev_open Module 40 determines the device to be opened from the vol_path parameters and calls the "volopen(vol_path)" driver module 42. Volopen module 42 tests for exclusive open status and tests for an exclusive open request received from the user's open volume operation 36. Volopen module 42 operates on the volume data 44 assigned to the device requested by the open request. Volopen module determines from the volume data if there are conflicts in open requests from

open volume operation 36 and from open device operation 38.

In Fig. 4 the open requests from open device operation 38 pass through "dev_open(fd_path)" driver module 46 to "vol_st_open(fd_path)" driver module 48. Dev_open Module 46 determines the device to be opened from the fd_path parameters and calls the "vol_st_open(fd_path)" driver module 48. Vol_st_open module 48 just like volopen module 42 operates on the volume data 44 to resolve conflicts in open requests from open volume operation 36 and from open device operation 38.

The structure of the volume data 44 is illustrated in Fig. 5 and the operation flow of volopen module 42 and vol_st_open module 48 are shown in Figures 6 and 7, respectively. In Fig. 5, the volume data 44 structure is shown expanded as storage areas 44a. The volume data comprises meta data that describes characteristics of the volume and device where the volume is located. The characteristics of interest to the preferred embodiment of the invention are open count 50 and exclusive flag 52. These characteristics are used by volopen module 42 and vol_st_open module 48 to resolve conflicts in access to a device.

In Fig. 6, the volopen process and logic operations for resolving conflicts are shown. When the volopen module is called, operation 53 reads the volume data characteristics from the volume meta-data for the requested device. Decision operation 54 first tests the volume data 44 for an exclusive flag for the device. If there is an exclusive flag, the process branches to operation 56, and operation 56 returns an EBUSY code to the volume management system. The volume management would then process the next user operation request. If there is no exclusive flag set for the requested device, the process branches to decision operation 58.

Decision operation 58 detects whether the volume open request from the user is an exclusive open request. If the open request is not exclusive, the process branches to increment operation 60 to increase the open count in the volume data by one. Operation 61 grants the open request, and a return success code goes to the operating system.

If the open request is exclusive, the processes branches YES from decision operation 58 to decision operation 62. Operation 62 tests whether the open count in the volume data characteristics is non-zero. If there is any count in the open count characteristic, operation 64 returns an EBUSY code to the volume management system. In this situation, the device is already open, so an exclusive open is not possible at this time. If the open count is zero, the process branches NO from decision operation 62 to set operation 66. Operation 66 sets the exclusive flag in the volume data for the requested device. Operation 60 then increments the open count characteristic, operation 61 grants the open request, and the volopen module operation for volume access process is complete.

If the user request was an open device request, the process or logical operations are performed by vol_st_open module as shown in Fig. 7. When the vol_st_open module is called, operation 67 reads the volume data characteristics for the volume data assigned to the requested device. Decision operation 68 detects whether the volume data 44 contains an exclusive flag for the device. If there is an exclusive flag, the process branches YES to return operation 70. Operation 70 returns an EBUSY code to the volume management system. If there is no exclusive flag set for the requested device, the process branches to decision operation 72.

Decision operation 72 tests if the device open request from the user is an exclusive open request. If the open request is not exclusive, the process branches to increment operation 74 to increase the open count in the volume data by one. Operation 75 grants the open request, and a success code is returned to the operating system.

If the open request is exclusive, decision operation 72 branches the process to decision operation 76. Operation 76 detects whether the open count in the volume data characteristics is non-zero. If there is a count in the open count characteristic for the device in the volume data, the process branches to operation 78 which returns an EBUSY code to the volume management system. If the open count is zero, the process branches NO from decision operation 76 to set exclusive flag operation 80. Operation 80 sets the exclusive flag for the requested device in the volume data, and the process flows to "clone me" operation 82.

Clone me operation 82 changes the identification of the device to the host using system. This causes the using system to set another open count characteristic for the device even though the device being accessed is unchanged. The purpose of this cloning operation which sets a new open count characteristic is to create the appearance of a new device open count for each open device request from the user. In the UNIX operating system, open device operations from the using system are sent for each open request. However if there are multiple accesses to the same device, the operating system only sends a close request once at the end of all the open operations, i.e. at the conclusion of the last access operation. Accordingly the volume management system when dealing with open device operations must have a mechanism to balance open requests and close requests. This is accomplished by the cloning operation 82 which creates a pseudo device identity and thus volume data for each open device request so that the operating system will send a close device request for each open device request when each access is completed. After the cloning operation 82, operation 74 increments the new open count characteristic for the pseudo device to one, and the vol_st_open operation is complete.

When the read or write access is completed for the requested volume or device, close operations are sent

by the user to the kernel of the UNIX operating system. The close operation is the same as the open operation as illustrated in Fig. 4 except the drive modules are close modules. Also, the volclose(vol_path) operation and vol_st_close(fd_path) are much simpler than volopen(vol_path) operation and the vol_st_open(fd_path) operation.

As illustrated in Fig. 8, the volclose operation and vol_st_close operation are identical. These processes only have to reset the open count characteristic and the exclusive flag characteristic of the volume data for the requested device. When either volclose or vol_st_close are called, the process clears the open count in operation 84 and clears the exclusive flag in operation 86. The volume management system is now ready to process further access requests to the device from user either by the volume access process or the device access process.

Claims

1. In a computing system, apparatus for managing the access to storage devices by user requests (36, 38) to the computing system, said apparatus comprising:

volume data (44) for each storage device stored in the computing system, said volume data including characteristics indicating the open status of each storage device;
 an open volume operative system (40) and an open device operative system (46) exchanging open status information for a storage device requested by a user through the volume data (44) for the requested storage device (32);
 said open volume operative system (40) responsive to the volume data characteristics for testing whether the requested device is already open exclusively and whether the open volume request (36) from the user application is an exclusive open request;
 said open device operative system (46) responsive to the volume data characteristics for testing whether the requested device is already open exclusively and whether the open device request (38) from the user application is an exclusive open request;
 said open volume operative system and said open device operative system return an open operation (42, 48) successful to permit read or write access to the device by the computing system when each operative system detects that the requested device is not already exclusively open or if the request is an exclusive open request that the device is not already open.

2. The apparatus of claim 1, and in addition:
 said open volume operative system and said open device operative system return a busy code (56) to the computing system when the requested storage device is already exclusively open (54) or when the requested storage device is already open and the request is for an exclusive open (58) of the storage device.
3. The apparatus of claim 1 or claim 2, wherein said volume data (44) includes an open indicator (50) and an exclusive open indicator (52) or exclusive flag for each storage device.
4. The apparatus of claim 3, wherein the open indicator is an open count (60) incremented for each open request and wherein:
 said open volume operative system (40) and said open device operative system (46) test for a non-zero count (62) to determine whether the requested storage device is already open.
5. The apparatus of claim 1, wherein the characteristics in the volume data for each storage device include an open count (50) indicating whether the storage device is open and an exclusive flag (52) indicating (54) whether the storage device is exclusively open.
6. The apparatus of claim 5 wherein:
 each of said open volume operative system and said open device operative system detect (62) a zero open count and a request for an exclusive open, and grant (61) the request, and set (66) an exclusive flag in the volume data for the requested storage device.
7. The apparatus of claim 6 wherein:
 each of said open volume operative system and said open device operative system detect a nonexclusive open request, grant the request, and increment (60) the open count for the requested storage device.
8. A method for managing access to storage devices by user requests to a computing system comprising the steps of:
 storing a volume data (44) characteristic list for each storage device, the characteristic list indicating the open status of each storage device: in response to an open volume request (36) or an open device request (38), detecting (54) from the characteristic list whether the requested device is already exclusively open; and in response to an open volume request or an open device request, detecting (58) whether a request for a storage device is an exclusive open request and testing (62) the characteristic list to detect if the requested device is already open, whereby the availability of the storage device is determined.
9. The method of claim 8, and in addition the steps of:
 returning to the computing system a busy code (56) if the requested device is already exclusively open; and
 returning a busy code (64) to the computing system if the request is an exclusive open request and the requested device is already open.
10. The method of claim 8 or claim 9, wherein the characteristic list includes an open count indicating the number of current open requests to each device and includes an exclusive flag indicating whether or not the device has been opened exclusively by a request.
11. The method of claim 10, and in addition the steps of:
 granting (61) an exclusive open request and setting the exclusive flag in the requested characteristic list if the requested device is not already open; and
 setting the open count to non-zero in the requested characteristic list if the open volume request or the open device request is granted, whereby the access to a storage device may be requested either as a volume request or a device request and conflicts between the requests are managed.

Patentansprüche

1. Vorrichtung in einem Computersystem für das Management des Zugriffs auf Speichergeräte durch Benutzeranforderungen (36, 38) an das Computersystem, wobei die Vorrichtung enthält:

Datenträgerdaten (44) für jedes Speichergerät, die in dem Computersystem gespeichert sind und Eigenschaften umfassen, die den geöffneten Status jedes Speichergeräts angeben; ein Datenträgeröffnung-Ausführungssystem (40) und ein Gerätöffnung-Ausführungssystem (46), die Informationen bezüglich des geöffneten Status für ein Speichergerät, das von einem Benutzer angefordert wird, anhand der Datenträgerdaten (44) für das angeforderte Speichergerät

- chergerät (32) austauschen;
wobei das Datenträgeröffnung-Ausführungssystem (40) auf die Datenträgerdaten-Eigenschaften anspricht, um zu prüfen, ob das angeforderte Gerät bereits exklusiv geöffnet ist und ob die Datenträgeröffnung-Anforderung (36) von der Benutzeranwendung eine exklusive Öffnungsanforderung ist;
wobei das Gerätöffnung-Ausführungssystem (46) auf die Datenträgerdaten-Eigenschaften anspricht, um zu prüfen, ob das angeforderte Gerät bereits exklusiv geöffnet ist und ob die Gerätöffnung-Anforderung (38) von der Benutzeranwendung eine exklusive Öffnungsanforderung ist;
wobei das Datenträgeröffnung-Ausführungssystem und das Gerätöffnung-Ausführungssystem eine Öffnungsoperation (42, 48) zurückleiten, die erfolgreich einen Leseoder Schreibzugriff auf das Gerät durch das Computersystem ermöglicht, wenn jedes Ausführungssystem erfaßt, daß das angeforderte Gerät nicht bereits ausschließlich geöffnet ist oder, falls die Anforderung eine exklusive Öffnungsanforderung ist, daß das Gerät nicht bereits geöffnet ist.
2. Vorrichtung nach Anspruch 1, wobei weiterhin:
- das Datenträgeröffnung-Ausführungssystem und das Gerätöffnung-Ausführungssystem an das Computersystem einen Belegcode (56) zurückleiten, wenn das angeforderte Speichergerät bereits exklusiv geöffnet ist (54) oder wenn das angeforderte Speichergerät bereits geöffnet ist und die Anforderung auf ein exklusives Öffnen (58) des Speichergeräts zielt.
3. Vorrichtung nach Anspruch 1 oder 2, wobei die Datenträgerdaten (44) einen Öffnungsanzeiger (50) und einen Exklusivöffnungsanzeiger (52) oder einen Exklusivmarker für jedes Speichergerät enthalten.
4. Vorrichtung nach Anspruch 3, wobei der Öffnungsanzeiger ein Öffnungszählstand (60) ist, der bei jeder Öffnungsanforderung inkrementiert wird, und wobei:
- das Datenträgeröffnung-Ausführungssystem (40) und das Gerätöffnung-Ausführungssystem (46) auf einen von null verschiedenen Zählstand (62) prüfen, um zu bestimmen, ob das angeforderte Speichergerät bereits geöffnet ist.
5. Vorrichtung nach Anspruch 1, wobei die Eigenschaften in den Datenträgerdaten für jedes Speichergerät einen Öffnungszähler (50), der angibt, ob das Speichergerät geöffnet ist, sowie einen Exklusivmarker (52), der angibt (54), ob das Speichergerät exklusiv geöffnet ist, enthalten.
6. Vorrichtung nach Anspruch 5, wobei:
- sowohl das Datenträgeröffnung-Ausführungssystem als auch das Gerätöffnung-Ausführungssystem einen Null-Öffnungszählstand und eine Anforderung für ein exklusives Öffnen erfassen (62), die Anforderung gewähren und in den Datenträgerdaten für das angeforderte Speichergerät einen Exklusivmarker setzen (66).
7. Vorrichtung nach Anspruch 6, wobei:
- sowohl das Datenträgeröffnung-Ausführungssystem als auch das Gerätöffnung-Ausführungssystem eine nicht exklusive Öffnungsanforderung erfassen, die Anforderung gewähren und den Öffnungszählstand für das angeforderte Speichergerät inkrementieren (60).
8. Verfahren für das Management des Zugriffs auf Speichergeräte durch Benutzeranforderungen an ein Computersystem, mit den folgenden Schritten:
- Speichern einer Eigenschaftsliste von Datenträgerdaten (44) für jedes Speichergerät, wobei die Eigenschaftsliste den geöffneten Status jedes Speichergeräts angibt;
als Antwort auf eine Datenträgeröffnung-Anforderung (36) oder eine Gerätöffnung-Anforderung (38) Erfassen (54), ob das angeforderte Gerät bereits exklusiv geöffnet ist, anhand der Eigenschaftsliste; und
als Antwort auf eine Datenträgeröffnung-Anforderung oder eine Gerätöffnung-Anforderung Erfassen (58), ob eine Anforderung für ein Speichergerät eine exklusive Öffnungsanforderung ist, und Prüfen (62) der Eigenschaftsliste, um zu erfassen, ob das angeforderte Gerät bereits geöffnet ist, wobei die Verfügbarkeit des Speichergeräts bestimmt wird.
9. Verfahren nach Anspruch 8, weiterhin mit den folgenden Schritten:
- Zurückleiten eines Belegcodes (56) an das Computersystem, falls das angeforderte Gerät bereits exklusiv geöffnet ist; und
Zurückleiten eines Belegcodes (56) an das Computersystem, falls die Anforderung eine Exklusivöffnungsanforderung ist und das angeforderte Gerät bereits geöffnet ist.

10. Verfahren nach Anspruch 8 oder Anspruch 9, wobei die Eigenschaftsliste einen Öffnungszählstand, der die Anzahl momentaner Öffnungsanforderungen für jedes Gerät angibt, und einen Exklusivmerker, der angibt, ob das Gerät durch eine Anforderung exklusiv geöffnet worden ist, enthält.

11. Verfahren nach Anspruch 10, weiterhin mit den folgenden Schritten:

Gewähren (61) einer Exklusivöffnungsanforderung und Setzen des Exklusivmerkers in der angeforderten Eigenschaftsliste, falls das angeforderte Gerät nicht bereits geöffnet ist; und Setzen des Öffnungszählers auf Nichtnull in der angeforderten Eigenschaftsliste, falls die Datenträgeröffnung-Anforderung oder die Geräteöffnung-Anforderung gewährt wird, wobei der Zugriff auf ein Speichergerät entweder als Datenträgeranforderung oder als Geräteanforderung angefordert werden kann und Konflikte zwischen den Anforderungen gemanagt werden.

Revendications

1. Dispositif, faisant partie d'un système informatique, destiné à gérer l'accès à des dispositifs de mémoire par des demandes d'utilisateurs (36, 38) adressées au système informatique, ledit dispositif comprenant:

des données de volume (44) pour chaque dispositif de mémoire, mémorisées dans le système informatique, lesdites données de volume comprenant des caractéristiques indiquant l'état ouvert de chaque dispositif de mémoire; un système (40) opérationnel d'ouverture de volume et un système (46) opérationnel d'ouverture de dispositif échangeant une information d'état d'ouverture pour un dispositif de mémoire demandé par un utilisateur par l'intermédiaire des données de volume (44) pour le dispositif de mémoire requis (32);

ledit système opérationnel d'ouverture de volume (40) réagissant aux caractéristiques de données de volume pour vérifier si le dispositif demandé est déjà ouvert de manière exclusive et si la demande d'ouverture de volume (38) provenant de l'application d'utilisateur est une demande d'ouverture exclusive;

ledit système opérationnel d'ouverture de dispositif (46) réagissant aux caractéristiques de données de volume pour vérifier si le dispositif demandé est déjà ouvert de manière exclusive et si la demande d'ouverture de dispositif (38) provenant de l'application d'utilisateur est une

demande d'ouverture exclusive;

ledit système opérationnel d'ouverture de volume et ledit système opérationnel d'ouverture de dispositif renvoient une opération d'ouverture (42, 48) réussie pour permettre l'accès en lecture ou en écriture au dispositif par le système informatique lorsque chaque système opérationnel détecte que le dispositif demandé n'est pas déjà ouvert de manière exclusive ou, si la demande est une demande d'ouverture exclusive, que le dispositif n'est pas déjà ouvert.

2. Dispositif selon la revendication 1, dans lequel, en outre:

ledit système opérationnel d'ouverture de volume et ledit système opérationnel d'ouverture de dispositif renvoient un code d'occupation (56) au système informatique lorsque le dispositif de mémoire demandé est déjà ouvert de manière exclusive (54) ou lorsque le dispositif de mémoire demandé est déjà ouvert et que la demande se rapporte à une ouverture exclusive (58) du dispositif de mémoire.

3. Dispositif selon la revendication 1 ou la revendication 2, dans lequel lesdites données de volume (44) comprennent un indicateur d'ouverture (50) et un indicateur d'ouverture exclusive (52), ou indicateur exclusif, pour chaque dispositif de mémoire.

4. Dispositif selon la revendication 3, dans lequel l'indicateur d'ouverture est un compteur d'ouverture (60) incrémenté à chaque demande d'ouverture et dans lequel:

ledit système opérationnel d'ouverture de volume (40) et ledit système opérationnel d'ouverture de dispositif (46) vérifient la présence d'une valeur non nulle (62) afin de déterminer si le dispositif de mémoire demandé est déjà ouvert.

5. Dispositif selon la revendication 1, dans lequel les caractéristiques dans les données de volume pour chaque dispositif de mémoire comprennent un compteur d'ouverture (50) indiquant si le dispositif de mémoire est ouvert et un indicateur exclusif (52) indiquant (54) si le dispositif de mémoire est ouvert de manière exclusive.

6. Dispositif selon la revendication 5, dans lequel:

chacun dudit système opérationnel d'ouverture de volume et dudit système opérationnel d'ouverture de dispositif détecte (62) un nombre d'ouverture nul et une demande d'ouverture exclusive, et accepte (61) la demande, et posi-

tionne (66) un indicateur exclusif dans les données de volume pour le dispositif de mémoire demandé.

7. Dispositif selon la revendication 6, dans lequel: 5

chacun dudit système opérationnel d'ouverture de volume et dudit système opérationnel d'ouverture de dispositif détecte une demande d'ouverture non exclusive, accepte la demande, et incrémente (60) le compteur d'ouverture pour le dispositif de mémoire demandé. 10

8. Procédé de gestion d'accès à des dispositifs de mémoire par des demandes d'utilisateur à un système informatique comprenant les étapes de: 15

mémorisation d'une liste caractéristique de données de volume (44) pour chaque dispositif de mémoire, la liste caractéristique indiquant l'état d'ouverture de chaque dispositif de mémoire; 20

en réponse à une demande d'ouverture de volume (36) ou une demande d'ouverture de dispositif (38), détection (54), à partir de la liste caractéristique, du fait que le dispositif demandé est déjà ouvert de manière exclusive; et 25

en réponse à une demande d'ouverture de volume ou une demande d'ouverture de dispositif, détection (58) du fait que la demande pour un dispositif de mémoire est une demande d'ouverture exclusive et vérification (62) de la liste caractéristique afin de détecter si le dispositif demandé est déjà ouvert, de sorte que la disponibilité du dispositif de mémoire est déterminée. 30 35

9. Procédé selon la revendication 8, comprenant, en outre, les étapes de: 40

renvoi, vers le système informatique, d'un code d'occupation (56) si le dispositif demandé est déjà ouvert de manière exclusive; et renvoi d'un code d'occupation (64) vers le système informatique si la demande est une demande d'ouverture exclusive et que le dispositif demandé est déjà ouvert. 45

10. Procédé selon la revendication 8 ou la revendication 9, dans lequel la liste caractéristique comprend un compteur d'ouverture indiquant le nombre de demandes d'ouvertures courantes pour chaque dispositif et comprend un indicateur exclusif indiquant si le dispositif a été ouvert de manière exclusive ou non par une demande. 50 55

11. Procédé selon la revendication 10, et comprenant, en outre, les étapes de:

acceptation (61) d'une demande d'ouverture exclusive et positionnement de l'indicateur exclusif dans la liste caractéristique demandée si le dispositif demandé n'est pas déjà ouvert; et positionnement du compteur d'ouverture à une valeur non nulle dans la liste caractéristique demandée si la demande d'ouverture de volume ou la demande d'ouverture de dispositif est acceptée, de sorte que l'accès à un dispositif de mémoire peut être demandé soit comme une demande de volume soit comme une demande de dispositif et les conflits entre les demandes sont gérés.

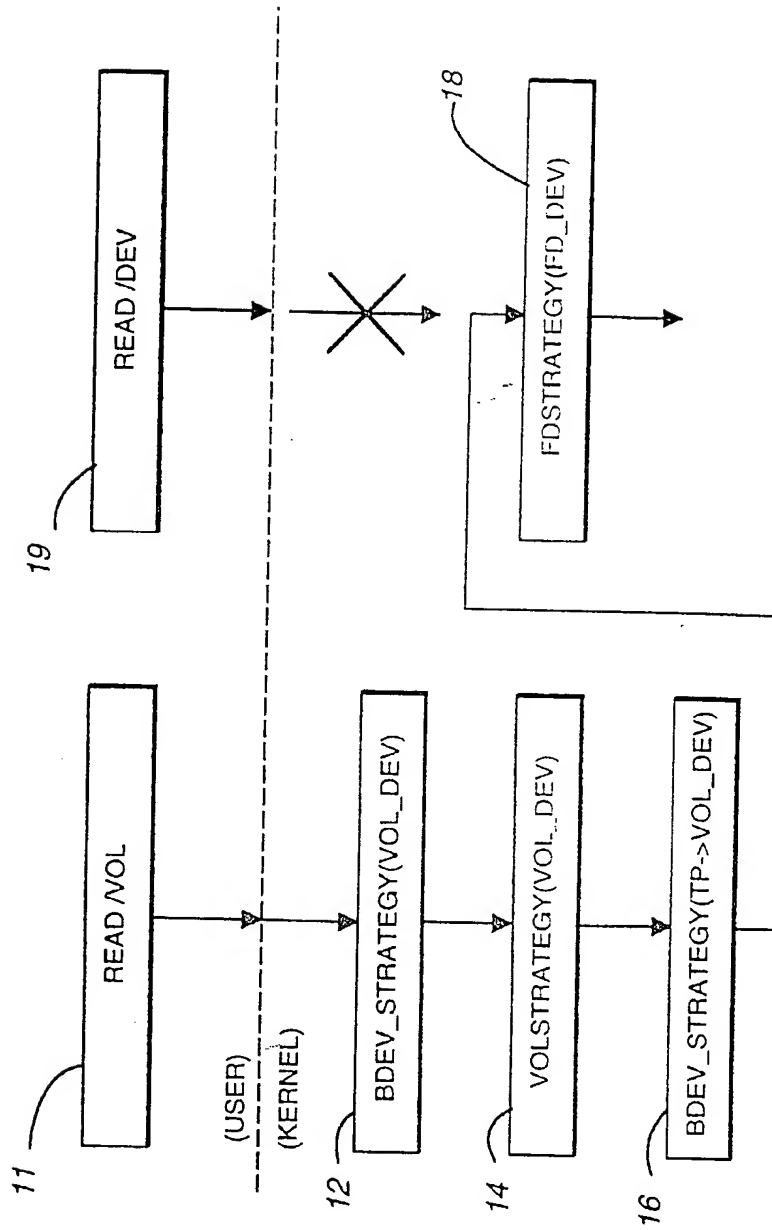


Fig. 1
Prior Art

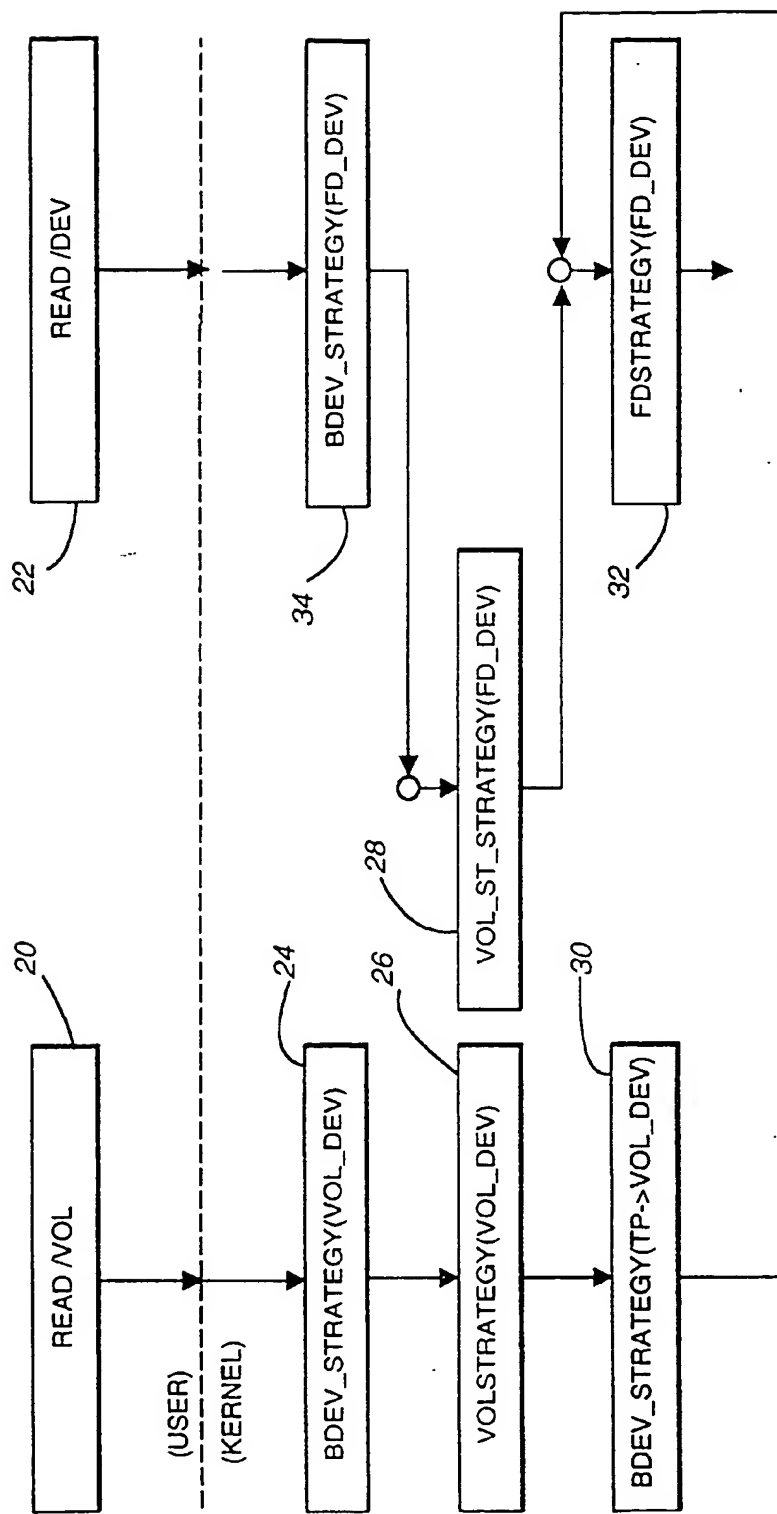


Fig. 2

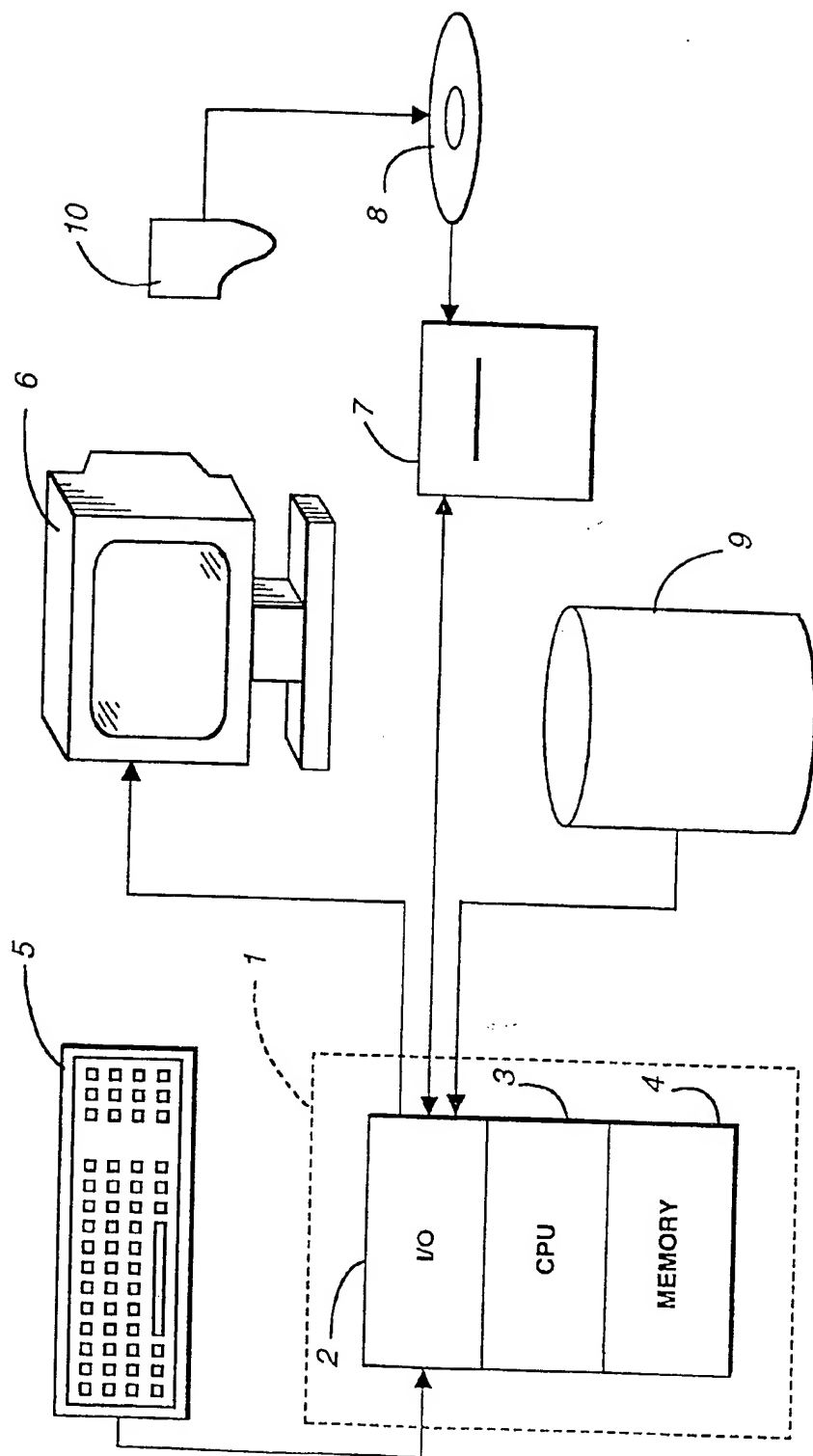


Fig. 3

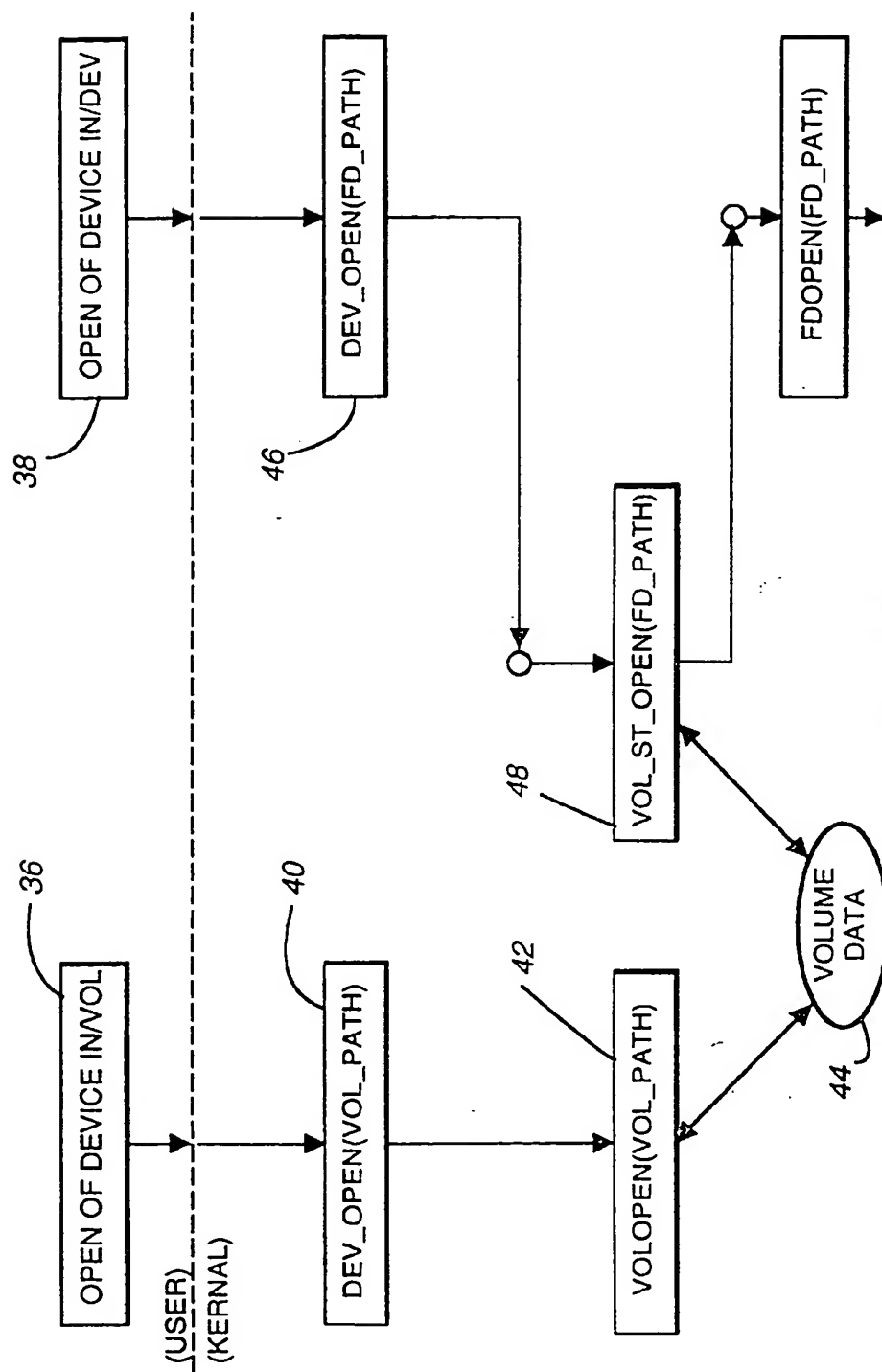


Fig. 4

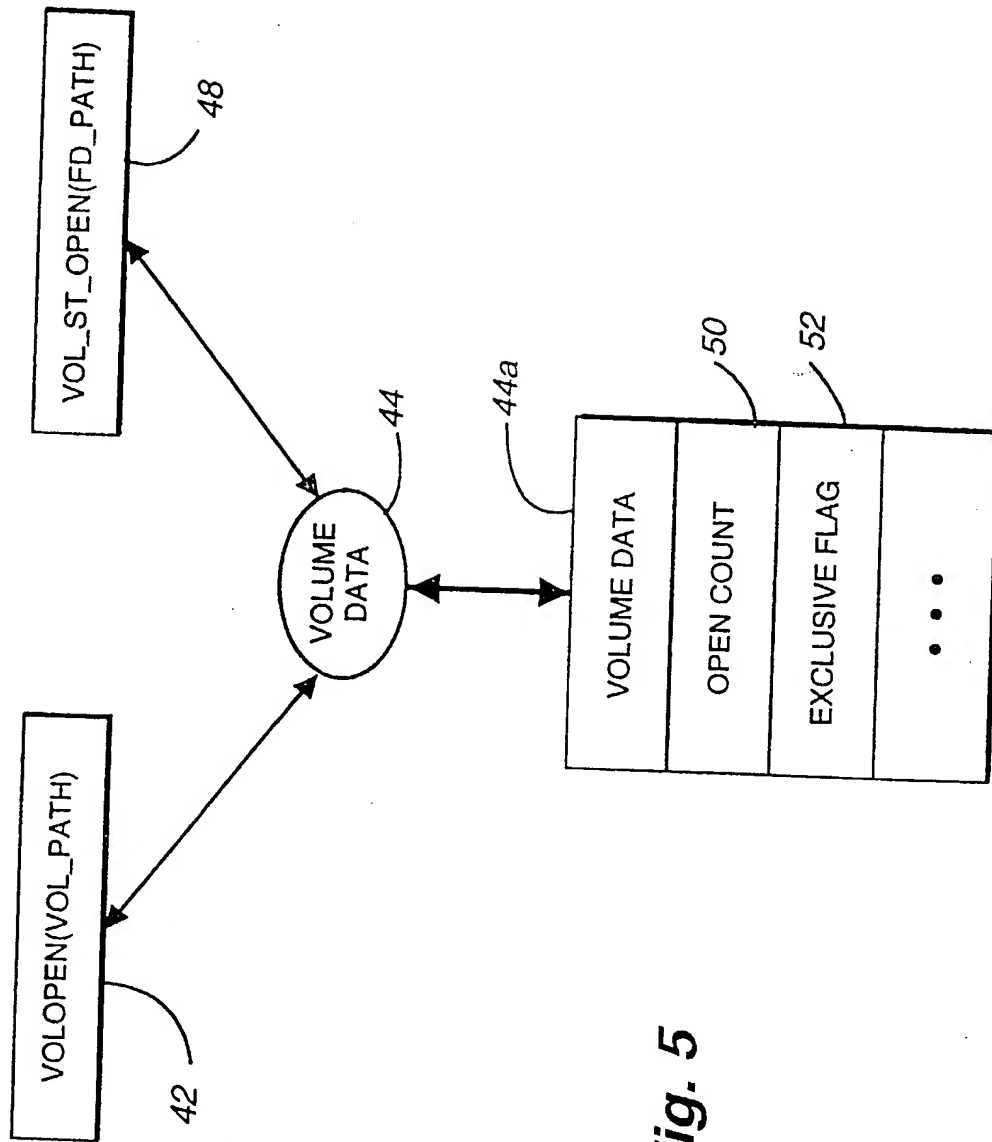
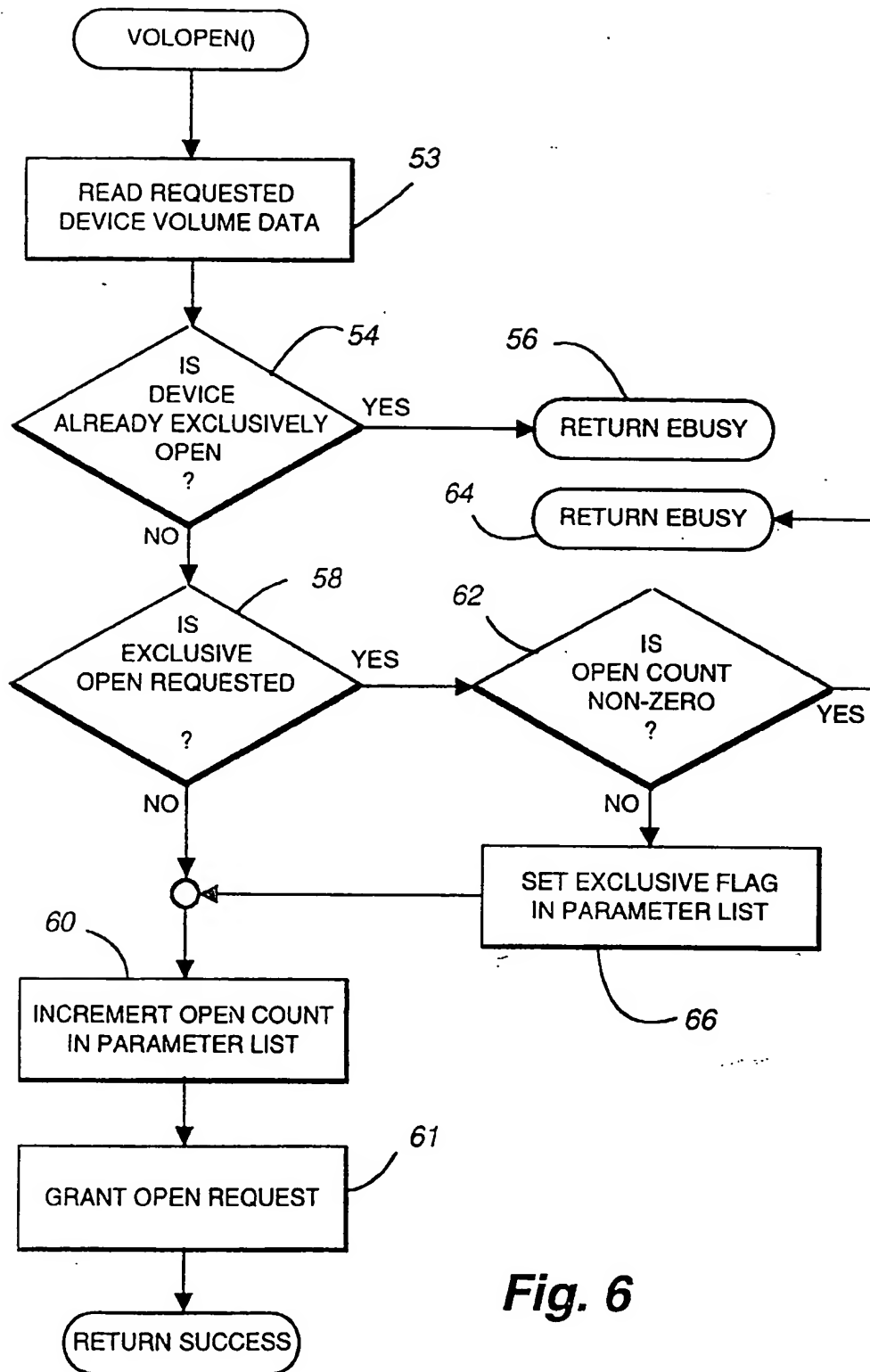
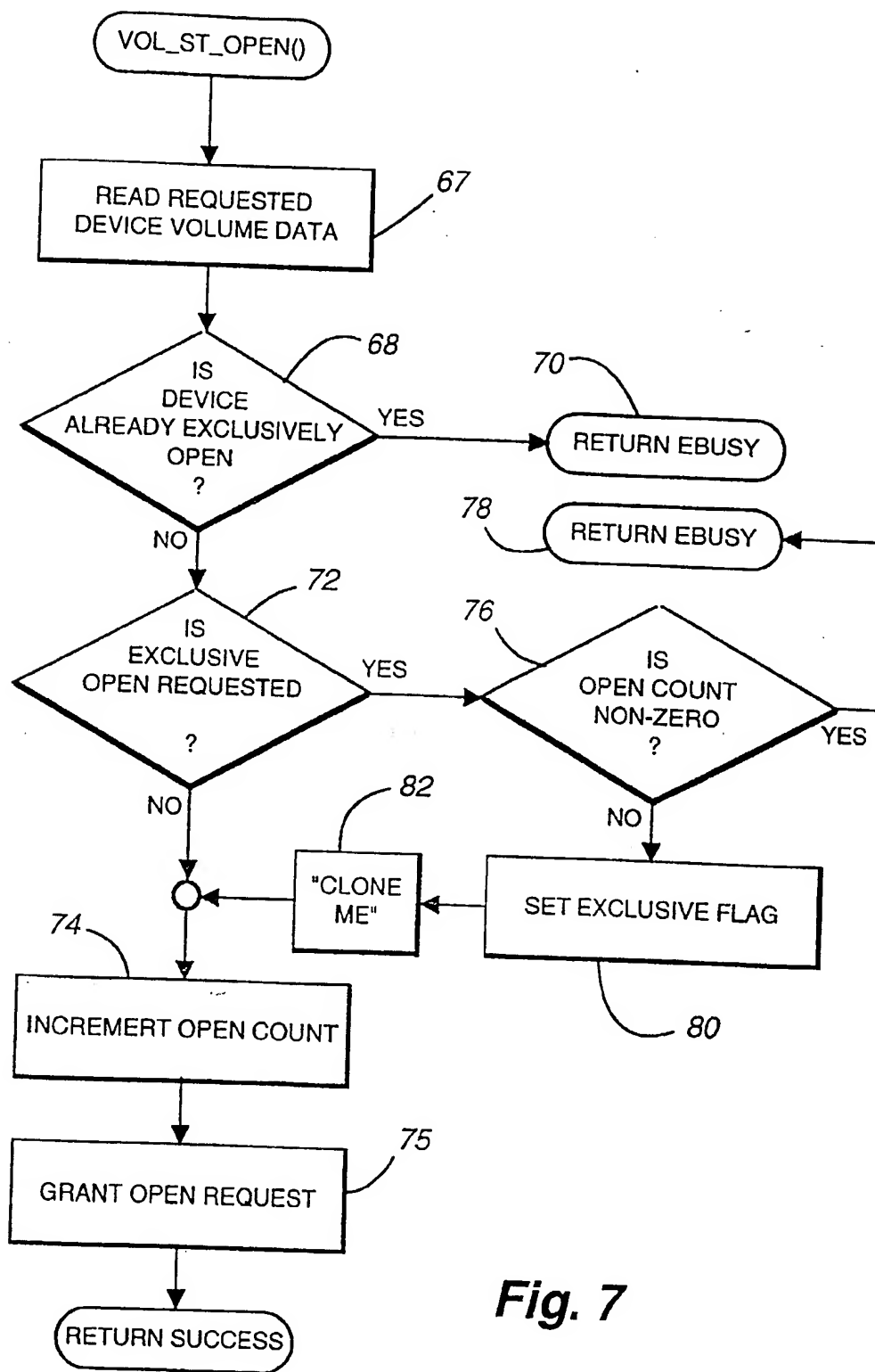


Fig. 5

**Fig. 6**

**Fig. 7**

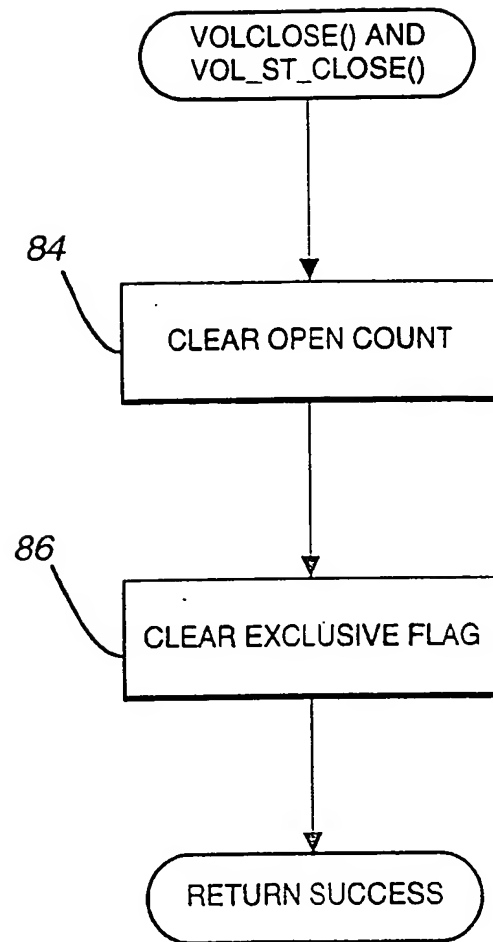


Fig. 8

THIS PAGE BLANK (USPTO)

RECEIVED
APR 30 2007
OIPF/JCWS